ARMY RESEARCH LABORATORY

Video Compression using Vector Quantization

by Mahesh Venkatraman, Heesung Kwon, and Nasser M. Nasrabadi

ARL-TR-1535 May 1998

Approved for public release; distribution unlimited.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-1535 **May 1998**

Video Compression using Vector Quantization

Mahesh Venkatraman, Heesung Kwon, and Nasser M. Nasrabadi
Sensors and Electron Devices Directorate

Approved for public release; distribution unlimited.

Abstract

This report presents some results and findings of our work on very-low-bit-rate video compression systems using vector quantization (VQ). We have identified multiscale segmentation and variable-rate coding as two important concepts whose effective use can lead to superior compression performance. Two VQ algorithms that attempt to use these two aspects are presented: one based on residual vector quantization and the other on quadtree vector quantization. Residual vector quantization is a successive approximation quantizer technique and is ideal for variable-rate coding. Quadtree vector quantization is inherently a multiscale coding method. The report presents the general theoretical formulation of these algorithms, as well as quantitative performance of sample implementations.

Contents

Figures

1. Introduction: Very-Low-Bit-Rate Video Coding for the Digital Battlefield

Battlefield digitization—the process of representing all components of a battlefield in digital form—allows the battlefield and its components to be visualized, simulated, and processed on computer systems, making the Army more deadly and reducing the use of physical resources. For complete battlefield digitization, images of various modalities must be gathered by different imaging techniques. These images are then processed to provide important information about the imaged areas.

An important class of visual data is the image sequence or video, and forward-looking infrared (FLIR) video is an important source of information. These data consist of a series of two-dimensional images captured at a constant temporal rate. The main drawback to effective use of this data source is the huge amount of raw digital data (bits) required to represent them. This volume of data makes real-time gathering and transmission over tactical internets impractical.

To effectively combat this problem, data compression is used: that is, techniques to reduce the number of bits required to represent the data. The large compression ratios needed to ”squeeze” video over low-bandwidth digital channels require the use of ”lossy” image compression techniques. Lossy compression techniques use a very small number of bits to represent the data at the cost of degraded information. These techniques require a trade-off between video quality and bit-rate constraints.

For intelligent compression of FLIR video images, the bit assignment should be made so that more bits are assigned to active areas, while fewer are assigned to passive background areas. Vector quantization (a block quantization technique) has this type of adaptability, so that it is highly suitable for compressing FLIR video.

In the work reported here, we systematically study the use of vector quantization for compressing video sequences. Results are shown for both FLIR video and regular gray-scale video. (A single representative frame of the original FLIR video scene and its compressed representation are shown in fig. 1.) We specifically study two adaptive vector quantization techniques: the residual vector quantizer (VQ) and the quadtree VQ. These two techniques permit the encoding of sources at different levels of precision depending on content.

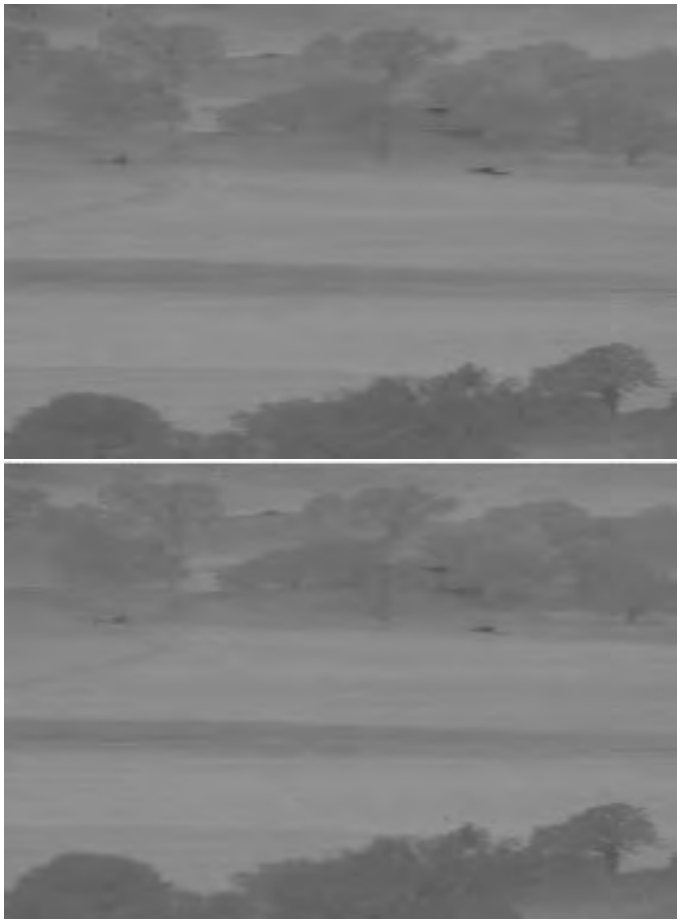

Figure 1. Original (top) and compressed (bottom) representations of a frame from a FLIR video sequence.

The targeted bit rate is in the very low (5 to 16 kb/s) range; this rate allows the compressed video to be transmitted over SINCGARS (Single-Channel Ground to Air Radio System) channels, as well as permitting multiple video streams to be multiplexed and transmitted over Fractional T1 lines. Such multiplex transmission would allow the video to be collected by sources such as unmanned airborne vehicles (UAVs) and transmitted to processing centers in real time.

2. Background

Images are represented in the digital domain by a matrix / array of intensity values, and video sequences are represented by a series of matrices. These matrices are often large, requiring large amounts of storage space and / or transmission bandwidth. When resources are limited (storage spaces or bandwidth), it is essential to reduce the amount of data necessary for representing the digital imagery. Data can be reduced (compressed) either with no loss in data (lossless compression) or with some degradation / distortion of the data (lossy compression). In lossless compression, redundancy in the data is removed, resulting in a smaller representation, but the ratio of compression that can be achieved is small. On the other hand, lossy compression techniques trade off the compression ratio against the tolerated distortion. Most current image- and video-compression techniques are lossy, since human visual perception can tolerate a certain amount of distortion in the presented visual data. Lossy compression can be achieved by *quantization*, a lossy compression technique in which data are represented at lower numerical precision than in the original representation.

2.1 Quantization

A quantization Q of a random variable $X \in \mathcal{R}$ is a mapping from $\mathcal{R}$ to $\mathcal{C}$, a finite subset of $\mathcal{R}$:

$$Q : \mathcal{R} \mapsto \mathcal{C}, \quad \mathcal{C} \subset \mathcal{R}. \tag{1}$$

The cardinality N_C of the set $\mathcal{C}$ gives the number of quantization levels. The mapping Q is generally a staircase function, as shown in figure 2, where $\mathcal{R}$ is divided into N_C segments $[b_i - 1, b_i), i = 1, \ldots, N$. Each $X_n \in [b_i - 1, b_i)$ is mapped to $c_i \in \mathcal{C}$, where c_i is the reconstruction value.

A sequence of random variables X_n can be quantized by two different methods. The first method involves each individual member of the sequence being quantized separately by the quantizer Q defined above. This method is called *scalar quantization*. In the second method, the sequence is grouped into blocks of adjacent members, and each block (a vector) is quantized by a *vector quantizer*. In the work reported here, vector quantization (sect. 3) is applied to video compression.

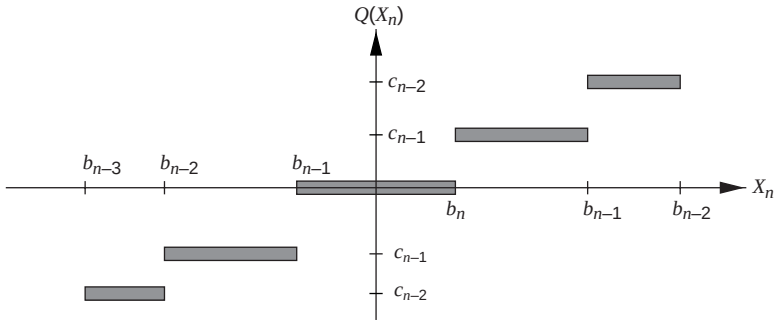

Figure 2. Scalar quantizer Q of a random variable X_n.

2.2 Video Compression

A video sequence is a three-dimensional signal of light intensity, with two spatial dimensions and a temporal dimension. A digital video sequence is a three-dimensional signal that is suitably sampled in all three dimensions; it is in the form of a three-dimensional matrix of intensity values. A typical video sequence has a significant amount of correlation between neighbors in all three dimensions. The type of correlation in the temporal dimension is significantly different from that in the spatial dimensions.

There are many different approaches to video compression, and some international compression standards have been established. Among the different approaches is a class of algorithms that first attempt to remove correlations in the temporal domain and then deal with removing correlations in the spatial dimensions. Among these is motion compensation (MC), a popular technique to remove the correlations in the temporal domain. Motion compensation results in a residue sequence, which is then quantized by two-dimensional quantization techniques similar to those used for compressing still images.

2.3 Motion Compensation

A video scene usually contains some motion of objects, occlusion / exposure of areas due to such motion, and some deformation. The rate of these changes is typically much smaller than the frame rate (i.e., the rate of sampling in the temporal dimension). Therefore, there is very little change between two adjacent frames. A motion-compensation algorithm exploits this consistency to approximate the current frame by using pieces from the previous frame. The result is a reasonable approximation of the current frame based on the previous one, with some side information in the form of motion vectors. The difference between the approximation of the current frame and the actual frame is quantized by a set of scalar or vector quantizers.

In figure 3, which shows the block diagram of the encoder and decoder, the difference between the approximation and the original is quantized by the quantizer Q. The encoder is a closed-loop system, as shown in the block diagram; it contains both a quantizer Q and an inverse quantizer Q^{-1}. Since the encoder uses the previous frame to approximate the current frame, the decoder needs the previous frame to generate the current frame. The decoder has only the quantized version of the previous frame and not the original frame. The inverse quantizer Q^{-1} in the encoder duplicates the decoder states at the encoder and gives the encoder access to a quantized version of the previous frame. The encoder uses this quantized version of the previous frame to generate an approximation of the current frame. This ensures that the approximation of the current frame generated from the previous frame is the same at both the encoder and decoder. Figure 4 shows the entropy of a sequence after (1) decorrelation by taking the frame difference and (2) decorrelation using block motion estimation. It can be seen that the entropy in this case is reduced by a factor of two compared to the original frame entropy. Between frames 105 and 120, the entropy of the residue due to motion estimation is significantly less than the entropy of the residue from frame differencing. This difference is due to significant motion of objects in the images during that period.

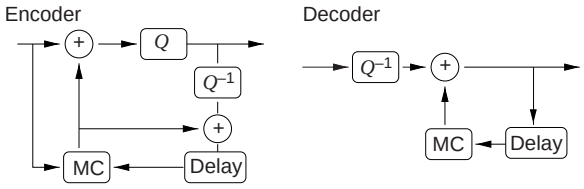

Figure 3. Motion compensation (MC) for video coding.

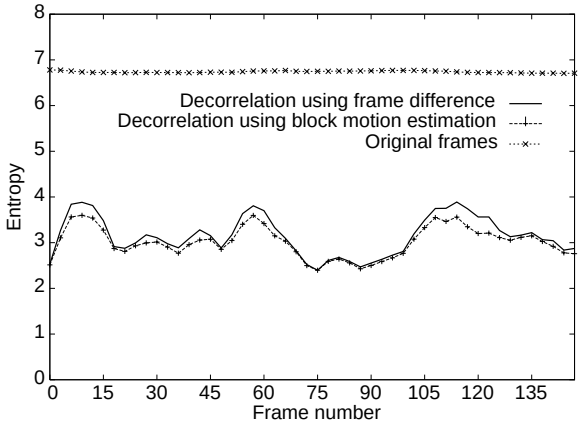

Figure 4. Entropy of a sequence after decorrelation in temporal dimension.

3. Vector Quantization

A vector quantizer Q is a mapping from a point in k-dimensional Euclidean space $\mathcal{R}^k$ into a finite subset $\mathcal{C}$ of $\mathcal{R}^k$ containing N reproduction points or vectors:

$$Q : \mathcal{R}^k \mapsto \mathcal{C}.$$

The N reproduction vectors are called *codevectors*, and the set $\mathcal{C}$ is called the *codebook*. $\mathcal{C} = (\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n)$, and $\mathbf{c}_i \in \mathcal{R}^k$ for each $i \in T = \{1, 2, \ldots, N\}$. The codebook $\mathcal{C}$ has N distinct members. The rate of the VQ, $r = \log_2(N)$, measures the number of bits required to index a member of the codebook.

A VQ partitions the space $\mathcal{R}^k$ into cells $\mathcal{R}_i$:

$$\mathcal{R}_i = \{\mathbf{X} \in \mathcal{R}^k : Q(\mathbf{X}) = \mathbf{c}_i\} \forall i \in T.$$

These cells represent the pre-image of the points $\mathbf{c}_i$ under the mapping Q, i.e., $\mathcal{R}_i = Q^{-1}(\mathbf{c}_i)$. These cells have the following properties:

$$\cup_i \mathcal{R}_i = \mathcal{R}^k,$$

$$\mathcal{R}_i \cap \mathcal{R}_j = \{\emptyset\} \forall i \neq j.$$

These properties imply that the cells are disjoint and that they cover the entire space $\mathcal{R}^k$.

The VQs dealt with here have the following additional properties:

- They are regular. The cells of a regular VQ, $\mathcal{R}_i$, are convex, and $\mathbf{c}_i \in \mathcal{R}_i$.
- They are polytopal. The cells of a polytopal VQ are polytopal. Polytopes are geometric regions bounded by hyperplane surfaces. A polytopal region is the intersection of a finite number of subspaces.
- They are bounded. A VQ is bounded if it is defined on a bounded domain $B \subset \mathcal{R}^k$; i.e., every input vector $\mathbf{X}$ lies in B.

A VQ consists of two operators, an encoder and a decoder. The encoder γ associates every input vector $\mathbf{X}$ to i, which is some member of the index set T. The decoder β associates the index i to $\mathbf{c}_i$, some member of the reproduction set $\mathcal{C}$:

$$\gamma : \mathcal{R}^k \mapsto T,$$
$$\beta : T \mapsto \mathcal{R}^k,$$
$$Q(\mathbf{X}) = \beta(\gamma(\mathbf{X})).$$

The block diagram of the VQ is shown in figure 5. The encoding operation is completely determined by the partition of the input space. The encoder identifies the cell to which a given input vector belongs. The decoding operation is determined by the codebook. Given the cell to which the input vector belongs, the decoder determines the reproduction vector that best represents the input vector. The decoder is very often in the form of a simple lookup table. Given the index, the table returns the vector entry corresponding to the index.

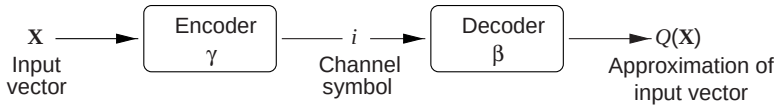

Figure 5. Encoder / decoder model of a VQ.

3.1 Quantization Error of Vector Quantizers

The performance of a VQ can be evaluated by the average distortion introduced by encoding a set of training input vectors. Ideally, the distortion should be zero. The output of the decoder should be a close representation of the input vector. The expected value of the distortion measure represents the performance of the quantizer:

$$\mathcal{D} = \mathbf{E}(\mathbf{d}(\mathbf{X}, Q(\mathbf{X}))),$$

where $\mathbf{d}(\mathbf{X}, Q(\mathbf{X}))$ represents the distortion introduced by the quantizer for the input vector $\mathbf{X}$.

One important distortion measure is the squared-error distortion measure (Euclidean distortion / L_2 distortion). This distortion measure is especially relevant to image coding problems, where the mean squared error is widely used as a quantitative measure of the performance of coding:

$$\begin{aligned} \mathbf{d}(\mathbf{X}, Q(\mathbf{X})) &= ||\mathbf{X} - Q(\mathbf{X})||^2, \\ \mathcal{D} &= \mathbf{E}(||\mathbf{X} - Q(\mathbf{X})||^2). \end{aligned}$$

Other distortion measures include the weighed squared-error distortion measure, the Mahalanobis distortion measure, and the Itakura-Saito distortion measure.

3.2 Optimality Conditions for Vector Quantizers

An optimal VQ is one that minimizes the overall distortion measure for any vector $\mathbf{X}$ with a probability distribution $P(\mathbf{X})$. A VQ has to satisfy two optimality conditions to achieve this minimum distortion:

- For a given fixed decoder β, the encoder γ should be the one that minimizes the overall distortion.
- For a given fixed encoder γ, the decoder β should be the best possible decoder.

3.2.1 Nearest-Neighbor Condition

Given a decoder, it is necessary to find the best possible encoder. The decoder contains a finite set of vectors $\mathcal{C}$, one of which is used to represent the input vector. For a given vector $\mathbf{X}$, the vector $\mathbf{c}_i$ is the nearest neighbor if

$$\mathbf{d}(\mathbf{X}, \mathbf{c}_i) \leq \mathbf{d}(\mathbf{X}, \mathbf{c}_j) \quad \forall\, \mathbf{c}_j \in \mathcal{C}.$$

The overall distortion for a given fixed codebook $\mathcal{C}$ is given by

$$\mathcal{D} = \mathbf{E}(\mathbf{d}(\mathbf{X}, Q(\mathbf{X}))),$$

$$\mathcal{D} = \int \mathbf{d}(\mathbf{X}, Q(\mathbf{X}))P(\mathbf{X})\, d\mathbf{X};$$

clearly,

$$\int \mathbf{d}(\mathbf{X}, Q(\mathbf{X}))P(\mathbf{X})\, d\mathbf{X} \geq \int \mathbf{d}(\mathbf{X}, \mathbf{c}_i)P(\mathbf{X})\, d\mathbf{X},$$

where $\mathbf{c}_i$ is the nearest neighbor of $\mathbf{X}$. Therefore, the best possible encoder for a given decoder is the nearest-neighbor encoder.

3.2.2 Centroid Condition

For a fixed encoder, it is necessary to find the reproduction codebook that minimizes the overall distortion. For a given cell $\mathcal{R}_i$, the centroid $\mathbf{c}_i$ is defined as

$$\mathcal{D}(\mathbf{X}, \mathbf{c}_i) \leq \mathcal{D}(\mathbf{X}, \mathbf{c})\ \forall\, \mathbf{X}, \mathbf{c} \in \mathcal{R}_i, \mathbf{c}_i \in \mathcal{R}_i.$$

For a given probability distribution, and for a given encoder, the overall distortion is given by

$$\mathcal{D} = \int \mathbf{d}(\mathbf{X}, Q(\mathbf{X}))P(\mathbf{X})\, d\mathbf{X},$$

$$\mathcal{D} = \sum_i \int_{\mathcal{R}_i} \mathbf{d}(\mathbf{X}, \mathbf{c})P(\mathbf{X})\, d\mathbf{X}.$$

Clearly,

$$\sum_i \int_{\mathcal{R}_i} \mathbf{d}(\mathbf{X}, \mathbf{c}) P(\mathbf{X})\, d\mathbf{X} \geq \sum_i \int_{\mathcal{R}_i} \mathbf{d}(\mathbf{X}, \mathbf{c}_i) P(\mathbf{X})\, d\mathbf{X}.$$

Therefore, for a given encoder, the optimum decoder is the centroid of the nearest-neighbor partitions.

Consider the set A of all possible partitions of the input vectors, and a collection $\mathcal{C}^o$ of all possible reproduction sets. The optimum VQ is the pair

$$(\{\mathcal{R}_i\}, \mathcal{C}); \{\mathcal{R}_i\} \in A \text{ and } \mathcal{C} \in \mathcal{C}^o,$$

such that $\mathcal{R}_i$ is the nearest neighbor partition of $\mathcal{C}$ that contains the centroids of the partitions in $\mathcal{R}_i$. These two conditions are generalizations of the Lloyd-Max conditions for scalar quantizers.

3.3 Design of Vector Quantizers

Design of VQs is a very difficult problem. For a given probability distribution $P(\mathbf{X})$, it is necessary to find the encoder and decoder that simultaneously satisfy both the nearest-neighbor condition and the centroid condition. Unfortunately, no closed-form solutions exist for even simple distributions.

A number of methods have been proposed for the design of VQs. All these are iterative methods based on finding the best VQ for a training set.

3.3.1 Generalized Lloyd's Algorithm

The generalized Lloyd's algorithm (GLA) (also known as the LBG algorithm after Linde, Buzo, and Gray [1]) is an iterative algorithm. This algorithm, which is similar to the k-means clustering algorithm, consists of two basic steps:

- For a given codebook $\mathcal{C}_t$, find the best partition $\{\mathcal{R}_i\}_t$ of the training set satisfying the nearest-neighbor neighborhood condition.
- For the new partition $\{\mathcal{R}_i\}_t$, find the best reproduction codebook $\mathcal{C}_{t+1}$ satisfying the centroid condition.

These two steps are repeated until the required codebook is obtained. The training algorithm begins with an initial codebook, which is refined by the Lloyd's iterations until an acceptable codebook is obtained. A codebook is considered acceptable if the error difference between the present and the previous codebooks is less than a threshold.

3.3.2 Kohonen's Self-Organizing Feature Map

Kohonen's self-organizing feature maps (KSOFMs) can be used to design VQs with optimal codebooks [2,3]. In this method of codebook design, an energy function for error is formulated and minimized iteratively. This design procedure is sequential, unlike GLA, which uses the batch method of training.

3.4 Entropy-Constrained Vector Quantizer

For transmission over a binary channel, the index of the reproduction vector from a codebook $\mathcal{C}$ of size N is represented by a binary string of length $\lceil \log_2 N \rceil$ bits. Often, it is possible to further reduce the number of bits required to represent the indices by using entropy coding as shown in figure 6. Entropy coding reduces the transmission entropy rate from $\lceil \log_2 N \rceil$ per block to almost the entropy rate of the index sequence. Since typical codebook design algorithms do not consider the possible entropy rates of the index sequences, the codebooks do not combine with an entropy coder in an optimal way. Design of entropy-constrained VQs (ECVQs) has been studied by Chou et al [4] (among others), who used a Lagrangian formulation with a gradient-based algorithm similar to the Lloyd's algorithm to design the codebooks.

Consider a vector $\mathbf{X} \in \mathcal{R}^k$ quantized by a VQ with a codebook $\mathcal{C} = \{\mathbf{c}_j : j = 1, \ldots, N\}$. Let $l(i)$ represent the length of the binary string used to represent the index i of the reproduction vector $\mathbf{c}_i$ of $\mathbf{X}$. Then the energy function that is minimized in the design of an ECVQ is given by [4]

$$J(\gamma, \beta) = \mathbf{E}[\mathbf{d}(\mathbf{x}_i, Q(\mathbf{x}_i)] + \lambda \mathbf{E}[l(i)], \tag{2}$$

where γ and β are the VQ encoder and decoder, respectively. The index entropy $\log(1/p(i))$ is used in the algorithm to represent the length of the binary string required to represent the index i. The codebook is then designed

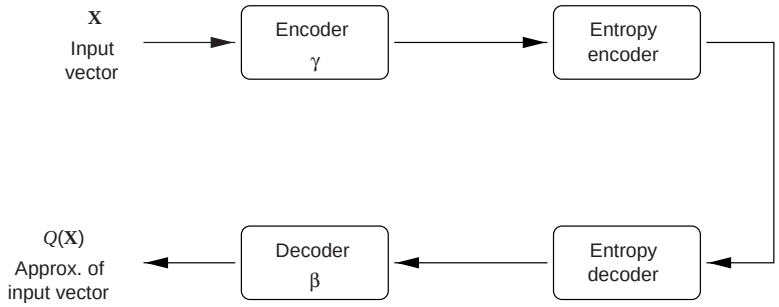

Figure 6. Entropy coding of VQ indices.

in an iterative manner, similar to the Lloyd's algorithm, through choosing an encoder and decoder that decrease the energy function in equation (2) at every iteration. Experimental results have shown that the ECVQ design algorithm described above gives an encoder-decoder pair that has superior numerical performance.

3.5 Video Compression Using Vector Quantization

The residual signal obtained after motion compensation can be compressed by vector quantization. The two-dimensional signal is divided into blocks of equal size, as shown in figure 7. The VQ encoder that is used to compress the residual signal is a nearest-neighbor encoder. It has a reference lookup table that contains the centroids of the VQ partitions. The encoder compares each block (in some predefined scanning order) with each member of the lookup table to find the closest match in terms of the defined distortion measure (usually the mean-squared error). The index of the closest matching codevector in the lookup table is then transmitted / stored as the compressed representation of the corresponding vector (block).

The decoder is a simple lookup table decoder, as shown in figure 8. The decoder uses the index symbol generated by the encoder as a reference to an entry in a lookup table in the decoder. The lookup table in the decoder is usually identical to the one in the encoder. This lookup table contains the possible approximations for the blocks in the reconstructed image. Based on the index, the approximate representation of the current block is determined. To generate the reconstructed two-dimensional array, the decoder places this representation at the position corresponding to the scanning order. The reconstructed array is used along with the motion-compensation algorithm to reproduce the compressed video sequence.

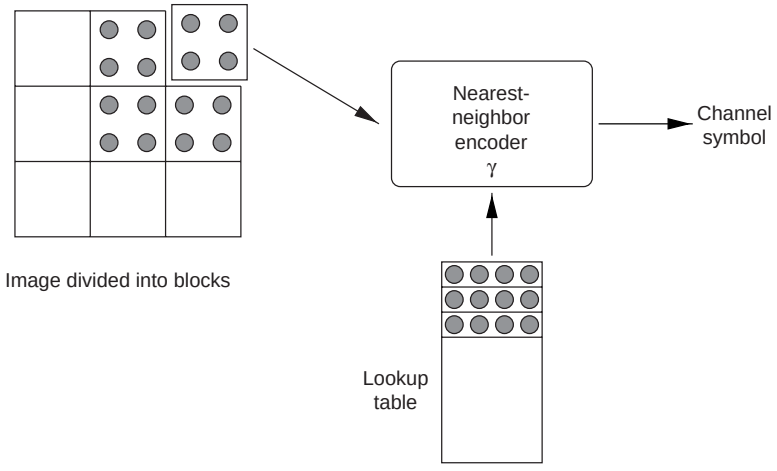

Figure 7. VQ encoder for two-dimensional arrays.

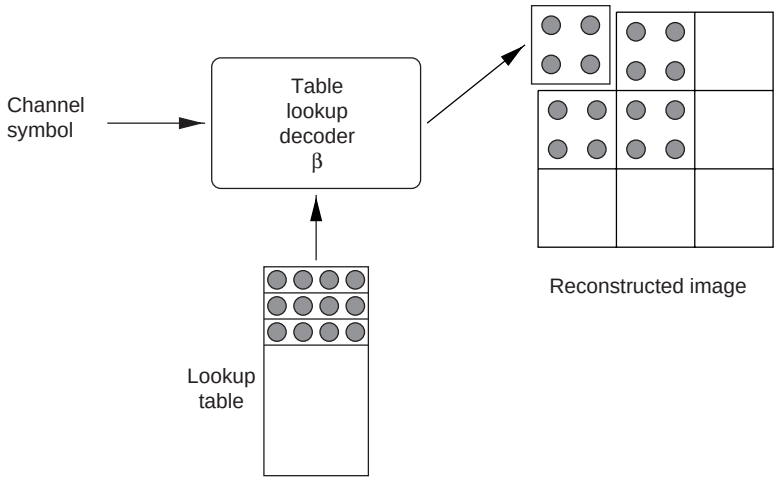

Figure 8. VQ decoder for two-dimensional arrays.

4. Residual Vector Quantization

Residual vector quantization (RVQ) is a structured vector quantization scheme proposed mainly to overcome the search and storage complexities of regular VQs [5,6]. Residual VQs are also known as multistage VQs. They consist of a number of cascaded VQs. Each stage has a VQ with a small codebook that quantizes the error signal from the previous stage. Residual quantizers are successive-refinement quantizers, where the information to be transmitted / stored is first approximated coarsely and then refined in the successive stages.

4.1 Residual Quantization

Consider a random variable X with a probability distribution function $P(X)$. Let $\mathbf{Q}^1(X)$ be an N^1 level quantizer and its associated bit rate be $\log_2(N^1)$ bits. The error due to this quantizer is

$$R^1 = X^1 - \mathbf{Q}^1(x^1),$$

and the expected value of the distortion is given by

$$\mathcal{D}^1 = \int \mathbf{d}(X^1, \mathbf{Q}^1(X^1))P(X^1)\, dX^1.$$

If the random variable needs to be represented more precisely (i.e., if the expected value of the distortion needs to be smaller), the first-stage residue can be quantized again by a second quantizer $\mathbf{Q}^2$. The quantizer $\mathbf{Q}^2$ approximates the random variable R^1. Let $\mathbf{Q}^2(R^1)$ be an N^2 level quantizer, and its associated bit rate be $\log_2(N^2)$ bits. The error due to this quantizer is given by

$$R^2 = R^1 - \mathbf{Q}^2(R^1),$$

and the expected value of distortion is now

$$\mathcal{D}^2 = \int \mathbf{d}[X, (\mathbf{Q}^1(X) + \mathbf{Q}^2(R^1))]P(X)\, dX.$$

This process can be thought of as a cascade of two quantizers, as shown in figure 9. The total bit rate of the quantization scheme is $\log_2(N^1)+\log_2(N^2)$.

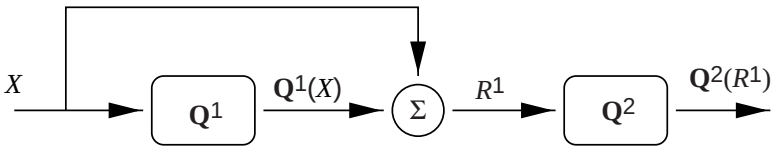

Figure 9. Residual quantizer—cascade of two quantizers.

This scheme can be extended to any number of quantizers. A K-stage residual quantizer consists of K quantizers $\{\mathbf{Q}^k : k = 1, \ldots, K\}$. Each quantizer $\mathbf{Q}^k$ quantizes the residue of the previous stage, $R^{(k-1)}$. The total bit rate of the quantization scheme is given by

$$B = \sum_{k=0}^{K} \log_2(N^k),$$

where N^k is the number of quantization levels of the quantizer $\mathbf{Q}^k$.

4.2 Residual Vector Quantizer

A residual VQ is a vector generalization of the residual quantizer outlined above. A K-stage residual VQ is composed of K VQs $\{\mathbf{Q}^k : k = 1, \ldots, K\}$. Each VQ consists of its own codebook $\mathcal{C}^k$ of size N^k. The kth-stage VQ operates on the residue $\mathbf{R}^{(k-1)}$ from the previous stage. The residue due to the first stage is given by

$$\mathbf{R}^1 = \mathbf{X} - \mathbf{Q}^1(\mathbf{X}).$$

The final quantized value of the vector $\mathbf{X}$ is given by

$$\mathbf{Q}(\mathbf{X}) = \mathbf{Q}^1(\mathbf{X}) + \mathbf{Q}^2(\mathbf{R}^1) + \ldots + \mathbf{Q}^k(\mathbf{R}^{(k-1)}) + \ldots + \mathbf{Q}^K(\mathbf{R}^{(K-1)}).$$

4.3 Search Techniques for Residual Vector Quantizers

The structure of a residual VQ inherently lends itself to a number of possible encoding schemes. Two of the main characteristics of encoding in residual quantizers are

- overall optimality—the least overall distortion at the end of the last stage of encoding, and
- stage-wise optimality—the least distortion possible at the end of each stage.

4.3.1 Exhaustive Search

Exhaustive search in residual quantizers aims at achieving the least overall distortion. In exhaustive search schemes, all possible combinations of all the stage quantizations are searched, and the combination giving rise to the least distortion is chosen. This search gives the best possible performance in the residual quantization scheme. But this search scheme is computationally very expensive and is the same as that of an unstructured VQ. The search complexity for a K-stage VQ with codebook sizes $\{N_1, N_2, \ldots N_K\}$ is of the order $O(N_1 \times N_2 \times \ldots N_K)$. Exhaustive search schemes are not particularly appropriate for progressive transmission schemes (successive refinement).

4.3.2 Sequential Search

Sequential search in residual quantizers makes full use of the structural constraint of the quantizer. The search process is stage by stage, wherein the quantization value that minimizes the distortion up to that stage is chosen. This search scheme is inherently inferior to exhaustive schemes, leading usually to suboptimal overall distortion performance. It is also the least expensive of all search schemes. The search complexity for a K-stage quantizer with codebook sizes $\{N_1, N_2, \ldots N_K\}$ is of the order $O(N_1 + N_2 + \ldots N_K)$. The search scheme is particularly well-suited for progressive transmission schemes.

4.3.3 M-Search

A hybrid search scheme, M-search, has been proposed [7] whose search complexity is less than that of a full search, but greater than that of a sequential. This scheme produces overall distortion performance that is better than that of sequential search schemes and close to that of the exhaustive search scheme. In this scheme, a subset of the quantization values is chosen at each stage based on the least distortion; these subsets are searched in an exhaustive fashion to get the quantized value.

4.4 Structure of Residual VQs

A quantizer partitions an input space into a finite number of polytopal regions (fig. 10). The centroid of each polytope approximates all the input symbols that belong to that particular region. The process of finding the residue of the signal is equivalent to shifting the coordinate system to the centroid of the polytope. This process is repeated for all the polytopes. Therefore, we have a finite set of spaces, each corresponding to a polytope. These spaces are bounded by the underlying polytope of the partition; i.e.,

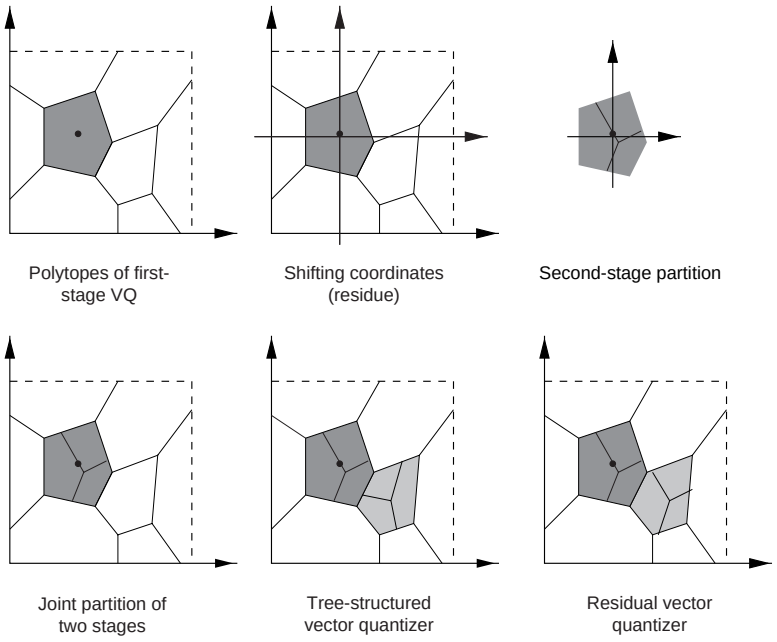

Figure 10. Structure of a residual VQ.

each of these spaces contains members around the origin that are limited in location by the polytope to which they belong.

Now consider the quantization of each of these spaces. If the optimal quantizer is found for each of these spaces, their structures (i.e., the partition of the input space) may be totally different. Such a quantizer is called a tree-structured quantizer. If a constraint is imposed such that the same partition structure is used for all the subspaces, then the method of quantization is the residual quantization scheme.

4.5 Optimality Conditions for Residual Quantizers

4.5.1 Overall Optimality

Consider a K-stage residual quantizer with a set of quantizers $\mathbf{Q}$,

$$\mathbf{Q} = \{Q^1, Q^2, \ldots, Q^k, \ldots Q^n\}$$

and codebooks $\mathbf{C}$,

$$\mathbf{C} = \{\mathcal{C}^1, \mathcal{C}^2, \ldots, \mathcal{C}^k, \ldots \mathcal{C}^n\},$$

with stage indices $\mathbf{K} = \{k : k = 1 \ldots K\}$. Each stage codebook $\mathcal{C}^k$ contains N^k codevectors $C^k = \{\mathbf{c}_1^k, \mathbf{c}_2^k, \ldots, \mathbf{c}_{N^k}^k\}$. As with VQs, we derive two optimality conditions. For the first condition, given the encoder, we find the best possible decoder. For the second condition, we find the best possible

encoder for a given decoder. We derive the conditions for a particular stage, assuming that all other stages have fixed encoders and decoders.

Centroid condition:
The index vector $\mathbf{I}$ belongs to the index space $\mathcal{I} = \{\mathbf{I} : \mathbf{I} = (i^1, i^2, \ldots, i^k, \ldots i^K), i^k = 1, \ldots, N^k\}$. Let the partition of the input space be $\mathcal{P}_\mathbf{I} = \mathcal{P}_{\mathbf{c}_{i^1}^1, \mathbf{c}_{i^2}^2, \ldots, \mathbf{c}_{i^k}^k, \ldots, \mathbf{c}_{i^K}^K}$, based on the different stage quantizers. This partition is based on a fixed encoder. To find the best decoder for the stage κ, let the decoders of stages $\{\mathbf{K}|\kappa\}$ be fixed. Overall distortion is given by

$$\mathcal{D}(\mathbf{X}, \mathbf{Q}(\mathbf{X})) = \sum_{\mathbf{I}\in\mathcal{I}} \sum_{\mathcal{P}_\mathbf{I}} (\mathbf{X} - \mathbf{c}_{i^1}^1 - \mathbf{c}_{i^2}^2 \ldots - \mathbf{c}_{i^{\kappa-1}}^{\kappa-1} - \mathbf{c}_{i^\kappa}^\kappa - \mathbf{c}_{i^{\kappa+1}}^{\kappa+1} \ldots - \mathbf{c}_{i^K}^K)^2 P(\mathbf{X}). \quad (3)$$

For the codevector $\mathbf{c}_{\iota^\kappa}^\kappa$ of the κth stage to be optimal, the following has to be true:

$$\frac{\delta\mathcal{D}(\mathbf{X}, \mathbf{Q}(\mathbf{X}))}{\delta\mathbf{c}_{\iota^\kappa}^\kappa} = 0; \quad (4)$$

that is,

$$\sum_{\mathbf{I}\in\mathcal{I}_{\iota^\kappa}} \sum_{\mathcal{P}_\mathbf{I}} (\mathbf{X} - \mathbf{c}_{i^1}^1 - \mathbf{c}_{i^2}^2 \ldots - -\mathbf{c}_{i^{\kappa-1}}^{\kappa-1} - \mathbf{c}_{\iota^\kappa}^\kappa - \mathbf{c}_{i^{\kappa+1}}^{\kappa+1} \ldots - \mathbf{c}_{i^K}^K) P(\mathbf{X}) = 0, \quad (5)$$

where $\mathcal{I}_{\iota_k} = \{\mathbf{I} \in \mathcal{I} : i_k = \iota_k\}$. Solving for $\mathbf{c}_{\iota^\kappa}^\kappa$, we get

$$\mathbf{c}_{\iota^\kappa}^\kappa = \frac{\sum_{\mathbf{I}\in\mathcal{I}_{\iota^\kappa}} \sum_{\mathcal{P}_\mathbf{I}} (\mathbf{X} - \mathbf{c}_{i^1}^1 - \mathbf{c}_{i^2}^2 \ldots - -\mathbf{c}_{i^{\kappa-1}}^{\kappa-1} - \mathbf{c}_{i^{\kappa+1}}^{\kappa+1} \ldots - \mathbf{c}_{i^K}^K) P(\mathbf{X})}{\sum_{\mathbf{I}\in\mathcal{I}_{\iota^\kappa}} \sum_{\mathcal{P}_\mathbf{I}} P(\mathbf{X})}. \quad (6)$$

A similar equation has been derived elsewhere [7]. This result has been described [7] as the centroid of the grafted residue.

Nearest-neighbor condition:
For fixed decoders and fixed encoders for stages $\mathbf{K}|\kappa$, the optimal stage κ encoder is one that either minimizes the overall distortion or minimizes the distortion for that stage. In either case, the mapping that produces the least distortion is the nearest-neighbor mapping. For exhaustive search decoders, the best encoder is the nearest-neighbor mapping for the direct-sum codebook.

4.5.2 Causal Stages Optimality

For the encoder to be optimal in terms of quantizers up to the present stage, the optimality conditions are as follows.

Centroid condition:
Let the partition of the input space be $\mathcal{P}'_\mathbf{I} = \mathcal{P}_{\mathbf{c}_{i^1}^1, \mathbf{c}_{i^2}^2, \ldots, \mathbf{c}_{i^\kappa}^\kappa}$, based on the

causal stage quantizers. For a fixed encoder, the optimal κth stage code is given by

$$\mathbf{c}^{\kappa}_{\iota^{\kappa}} = \frac{\sum_{\mathbf{I}\in\mathcal{I}'_{\iota^{\kappa}}}\sum_{\mathcal{P}'_{\mathbf{I}}}(\mathbf{X}-\mathbf{c}^{1}_{i^1}-\mathbf{c}^{2}_{i^2}\ldots--\mathbf{c}^{\kappa-1}_{i^{\kappa-1}}-\mathbf{c}^{\kappa+1}_{i^{\kappa+1}}\ldots-\mathbf{c}^{K}_{i^K})P(\mathbf{X})}{\sum_{\mathbf{I}\in\mathcal{I}'_{\iota^{\kappa}}}\sum_{\mathcal{P}'_{\mathbf{I}}}P(\mathbf{X})}, \quad (7)$$

where $\mathcal{I}'_{\iota^{\kappa}} = \{\mathbf{I} : \mathbf{I} = (i^1, i^2, \ldots, i^k, \ldots i^{\kappa}), i^k = 1, \ldots, N^k\}$ is the index vector of the causal stages including the present stage; $\mathbf{c}^{\kappa}_{\iota^{\kappa}}$ is the centroid of the direct partition up to the κ stage.

Nearest-neighbor condition:
For fixed decoders and fixed encoders for stages $\mathbf{K}|\kappa$, the optimal κth stage encoder is the nearest-neighbor mapping encoder.

4.5.3 Simultaneous Causal and Overall Optimality

Consider the κth stage encoder of a $\mathbf{K}$-stage residual VQ. Let us assume that it is optimal in terms of both causal and overall distortion. Then it satisfies the following two equations simultaneously:

$$\mathbf{c}^{\kappa}_{\iota^{\kappa}} = \frac{\sum_{\mathbf{I}\in\mathcal{I}_{\iota^{\kappa}}}\sum_{\mathcal{P}_{\mathbf{I}}}(\mathbf{X}-\mathbf{c}^{1}_{i^1}-\mathbf{c}^{2}_{i^2}\ldots--\mathbf{c}^{\kappa-1}_{i^{\kappa-1}}-\mathbf{c}^{\kappa+1}_{i^{\kappa+1}}\ldots-\mathbf{c}^{K}_{i^K})P(\mathbf{X})}{\sum_{\mathbf{I}\in\mathcal{I}_{\iota^{\kappa}}}\sum_{\mathcal{P}_{\mathbf{I}}}P(\mathbf{X})}, \quad (8)$$

$$\mathbf{c}^{\kappa}_{\iota^{\kappa}} = \frac{\sum_{\mathbf{I}\in\mathcal{I}'_{\iota^{\kappa}}}\sum_{\mathcal{P}'_{\mathbf{I}}}(\mathbf{X}-\mathbf{c}^{1}_{i^1}-\mathbf{c}^{2}_{i^2}\ldots--\mathbf{c}^{\kappa-1}_{i^{\kappa-1}})P(\mathbf{X})}{\sum_{\mathbf{I}\in\mathcal{I}'_{\iota^{\kappa}}}\sum_{\mathcal{P}'_{\mathbf{I}}}P(\mathbf{X})}. \quad (9)$$

The two denominators are equal; therefore,

$$\sum_{\mathbf{I}\in\mathcal{I}_{\iota^{\kappa}}}\sum_{\mathcal{P}_{\mathbf{I}}}(\mathbf{X}-\mathbf{c}^{1}_{i^1}-\mathbf{c}^{2}_{i^2}\ldots--\mathbf{c}^{\kappa-1}_{i^{\kappa-1}}-\mathbf{c}^{\kappa+1}_{i^{\kappa+1}}\ldots-\mathbf{c}^{K}_{i^K})P(\mathbf{X}) = \sum_{\mathbf{I}\in\mathcal{I}'_{\iota^{\kappa}}}\sum_{\mathcal{P}'_{\mathbf{I}}}(\mathbf{X}-\mathbf{c}^{1}_{i^1}-\mathbf{c}^{2}_{i^2}\ldots--\mathbf{c}^{\kappa-1}_{i^{\kappa-1}})P(\mathbf{X}). \quad (10)$$

Simplification of equation (10) gives

$$\sum_{\mathbf{I}\in\mathcal{I}_{\iota^{\kappa}}}\sum_{\mathcal{P}_{\mathbf{I}}}(\mathbf{c}^{\kappa+1}_{i^{\kappa+1}}\ldots-\mathbf{c}^{K}_{i^K})P(\mathbf{X}) = 0. \quad (11)$$

Basically, for the encoder to have simultaneous global and stage-wise optimality at any given stage κ, the sum of the codevectors of stages $\kappa+1\ldots\mathbf{K}$ must equal zero. This suggests that a successive-refinement residual VQ is not optimal. A rigorous treatment of successive approximation is given by Equitz and Cover [8].

4.6 Design of Residual Vector Quantizers

The design of residual VQs is based on a number of trade-offs, and different training methods are used for different coding schemes [9]. Sequential search quantizer codebooks in general are different from exhaustive search quantizer codebooks. A number of design methods have been proposed. Gupta et al [10] have proposed a joint codebook design in which all but one stage is fixed, and one particular stage codebook is adapted to minimize overall distortion. During the next step of the iteration, another codebook is adapted; this step is repeated cyclically until the required convergence is obtained. Barnes and Frost [7] use a similar algorithm for codebook design. Rizvi and Nasrabadi [11] have proposed a design algorithm based on the Kohonen network, where an energy is iteratively minimized to reduce the overall distortion.

4.7 Residual Vector Quantization with Variable Block Size

The residual VQ outlined thus far quantizes blocks (vectors) of the same size at each stage. When an input sequence to be quantized contains non-stationary artifacts, it is often difficult to compress with fixed-block-size quantizers. Blocks containing discontinuities are quantized rather poorly by all the stages, or they require a large number of residual VQ stages. One way to solve the problem is to use smaller block sizes at the later stages of the residual VQ. A variable-block-size residual VQ is shown in figure 11. In this figure, the first-stage quantizer uses blocks of size 4, the second-stage quantizer uses blocks of size 2, and the third-stage quantizer uses blocks of size 1 (the third-stage quantizer is a scalar quantizer in this example).

4.8 Pruned Variable-Block-Size Residual Vector Quantizer

Often not every part of a digital signal needs to be quantized by all the stages of the residual VQ. Sections of the signal containing little or no information can easily be represented by just one or two stages. Restricting the number of stages for a particular section of the signal is equivalent to pruning the tree structure, as shown in figure 11. There are a number of different ways of pruning the tree. One significant characteristic of a pruned

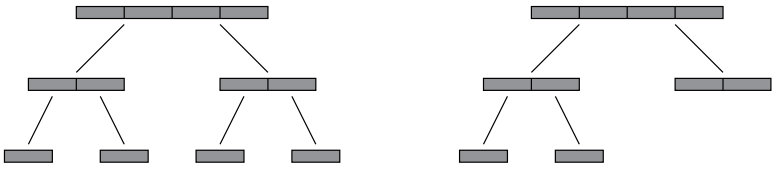

Figure 11. Tree structure of variable-block-size residual VQ. Pruning of right tree corresponds to variable-rate / variable-block-size residual VQ.

variable-block-size residual VQ is that very-low-bit-rate side information needs to be stored / transmitted. This side information determines the particular tree structure used for every block. The tree structure is required by the decoder, as it needs to know the number of stages used for every part of the sequence.

4.8.1 Top-Down Pruning Using a Predefined Threshold

This algorithm can decide the number of stages to be used for a particular block by examining the error after every stage. If the error at the end of a particular stage is less than a threshold, then the quantization can be stopped at that stage. The error is measured by a significance measure, such as the L_2 norm (mean squared error). If the L_2 distance after stage κ is less than a threshold τ_κ, then quantization is stopped at that stage. Choice of the threshold determines the performance of this algorithm. When a single global threshold $\tau = \tau_1 = \ldots = \tau_k = \ldots = \tau_K$ is used, it directly controls the output bit rate of the quantizer.

4.8.2 Optimal Pruning in the Rate-Distortion Sense

Optimal pruning in the rate-distortion sense is a bottom-up pruning technique in which a given block is quantized by all the stages. The quantization error is measured after every stage and stored. For tree pruning, the number of bits required to encode to a particular depth is traded off against the distortion. Let $\mathcal{T}$ represent a set of all possible tree structures and $\mathbf{Q}_T(\mathbf{X})$ be the quantized value of $\mathbf{X}$ corresponding to the tree structure T. Let $L(T)$ represent the number of bits required to represent a particular tree structure. For a particular tree structure T, let $\mathcal{I}_T$ represent the set of indices after quantization (based on the tree structure) and $L(\mathcal{I}_T)$ represent the number of bits required to encode the indices. A Lagrangian formulation can be made, and the tree can be pruned according to this objective function. This technique is equivalent to finding a particular tree structure that minimizes the following cost function:

$$\mathbf{d}(\mathbf{X}, \mathbf{Q}(\mathbf{X})) + \lambda \cdot (L(\mathcal{I}_T) + L(T)). \tag{12}$$

The value of the Lagrangian multiplier λ controls the output bit rate of the quantizer. It controls the slope of the tangent to the $\mathbf{R}$-$\mathbf{D}$ curve of the quantizer at different operating points, as shown in figure 12.

4.9 Transform-Domain Vector Quantization for Large Blocks

Direct vector quantization of large blocks is computationally expensive, and the design of the codebooks for large-block VQs is difficult. The complexity of VQs can be reduced through transform vector quantization [12].

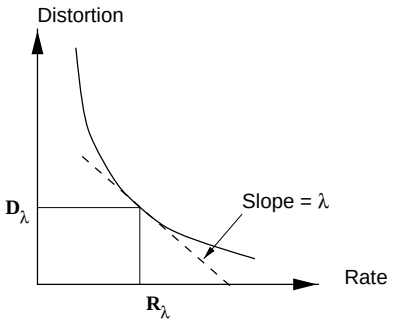

Figure 12. Optimal pruning of residual VQ in rate-distortion sense.

The data vector is first transformed by a decorrelating transformation Φ, such as the discrete cosine transform (DCT). A masking function M is then applied to the transformed data to reduce the dimensionality of the vector. In its simplest form, the masking function is a binary vector, and it truncates the number of coefficients used. The masking function can also contain a normalizing factor for each coefficient based on its variance. The resulting vector is then quantized by a small-vector-dimension quantizer. For decoding, the inverse of the mask function M^{-1} is first applied (usually in the form of padding with zeros), followed by the inverse transform Φ^{-1}. Use of a unitary transform like the DCT, which compacts the signal energy to a relatively small number of coefficients, leads to the requirement of a VQ with much smaller dimensions. It is therefore possible to use transform VQs in the initial stages of a variable-block-size residual VQ.

4.10 Video Compression Using Residual Vector Quantization

4.10.1 Theory of Residual Vector Quantization

The residual signal generated by the motion compensation algorithm, as described in section 2.3, can be compressed by a residual VQ. This section describes a particular implementation of an encoder using residual vector quantization. In the first two stages of the encoder, quantization is performed in the transform domain, as shown in figure 13. The residual signal $\mathbf{r}^0(i,j,t)$ is broken into blocks of dimension $m_1 \times n_1$, represented by $\mathbf{R}^0(I,J,t)$. The algorithm measures variances of each block and compares them to a threshold, to determine if the block needs to be encoded. Often, the background areas contain no information, since the motion estimation algorithm predicts the data perfectly. The vectors $\mathbf{R}^0(I,J,t)$, which require transmission, are then transformed to produce the transform-domain signal $\mathcal{R}^0(I,J,t)$, through a transform operator Φ:

$$\mathcal{R}^0(I,J,t) = \Phi[\mathbf{R}^0(I,J,t)]. \tag{13}$$

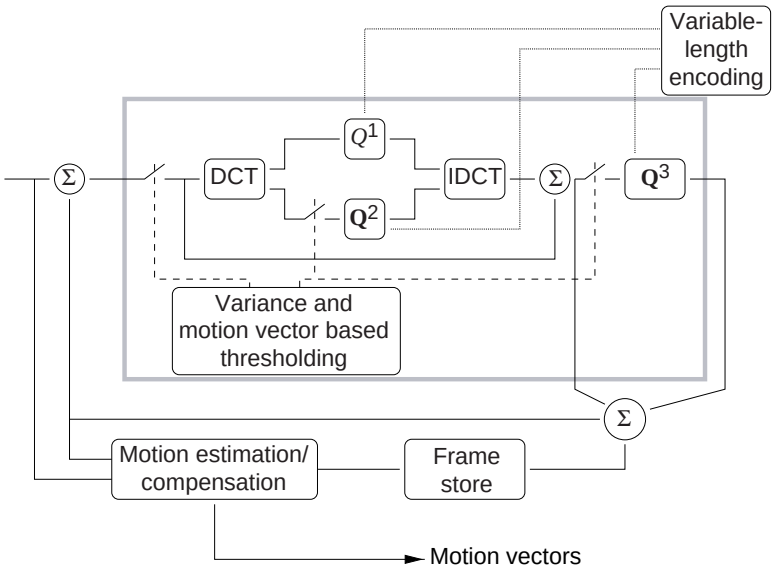

Figure 13. Video encoder based on residual vector quantization.

A masking operator M_1 is then applied to the transformed vector to produce a truncated vector of reduced dimension, $\mathcal{R}_m^0(I,J,t)$. This vector $\mathcal{R}_m^0(I,J,t)$ is quantized by the first-stage VQ. If $\mathbf{c}_{\iota^1}^1$ is the best matching code vector, the approximation of $\mathbf{R}^0(I,J,t)$ is given by

$$\mathbf{Q}_1[\mathbf{R}^0(I,J,t)] = \Phi^{-1}[M_1^{-1}(\mathbf{c}_{\iota^1}^1)], \tag{14}$$

where M_1^{-1} and Φ^{-1} are the inverses of the masking function and the transform operator. The error after the first-stage quantization is given by

$$\mathbf{R}^1(I,J,t) = \mathbf{R}^0(I,J,t) - \mathbf{Q}_1[\mathbf{R}^0(I,J,t)]. \tag{15}$$

This residual vector is measured for significance, and if it requires further compression, a second mask M_2 is applied to the transformed vector to produce vector $\mathcal{R}_m^1(I,J,t)$. This vector is quantized by the second-stage VQ. If the best match codevector is $\mathbf{c}_{\iota^2}^2$, the approximation of $\mathbf{R}^0(I,J,t)$ after the second stage is given by

$$\mathbf{Q}_2[\mathbf{R}^1(I,J,t)] = \Phi^{-1}[M_1^{-1}(\mathbf{c}_{\iota^1}^1) + M_2^{-1}(\mathbf{c}_{\iota^2}^2)]. \tag{16}$$

The masking functions M_1 and M_2 are binary templates, which together select the first few perceptually significant coefficients. The masking function effectively creates a low-pass-filtered version of the signal by discarding the higher frequency coefficients. The residue after the second stage is given by

$$\mathbf{R}^2(I,J,t) = \mathbf{R}^0(I,J,t) - \mathbf{Q}_2[\mathbf{R}^1(I,J,t)]. \tag{17}$$

This residual vector is then split into smaller blocks $\mathbf{R}'_2(I', J', t)$ of dimension $m_2 \times n_2$ for quantization by the subsequent stages. The algorithm compares the vectors $\mathbf{R}'_2(I', J', t)$ with a threshold to determine if they are significant enough to require transmission. The significant blocks that require transmission are quantized by the second-stage VQ, which gives an approximation $\mathbf{Q}_3[\mathbf{R}^2(I_2, J_2, t)]$. The process of decomposition into smaller blocks and selective quantization is applied recursively in the later stages, so that good representation is obtained of the residual signal $r_0(i, j, t)$. The indices of the quantizers are entropy coded by adaptive arithmetic coding [13].

The block diagram in figure 14 shows that operation of the decoder is not complex. The variable-length decoder recovers the bit maps and the code-vector indices from the arithmetically encoded sequence. The decoders are constructed with lookup tables. Based on complexity requirements, the lookup tables of the first two stages can store either the transform-domain coefficients $c^1_{\iota^1}$ and $c^2_{\iota^2}$ or the reconstruction vectors $\Phi^{-1}[M_1^{-1}(c^1_{\iota^1})]$ and $\Phi^{-1}[M_2^{-1}(c^2_{\iota^2})]$. In the first case, two additional operations—a padding operation (M^{-1}) and an inverse transform operation (Φ^{-1})—must be performed. In the second case, memory requirements are significantly larger. If the later-stage vectors are required, a direct table lookup is performed, using the indices, and the low-dimension vector is added to the first-stage reconstructed vector in the appropriate position. This process provides the reconstructed residual signal $\hat{\mathbf{R}}^0(I, J, t)$, which is then passed to the motion-compensation stage to produce the reconstructed frame.

In order to achieve a true variable rate, the encoder makes decisions about the number of quantization stages required by each stage. These decisions have to be transmitted to the decoder for the encoded data to be decoded correctly. The decisions are usually encoded as bit maps for each stage. For a three-stage encoder, three bit maps are required for proper decoding. These bit maps could require a significant portion of the bit budget if

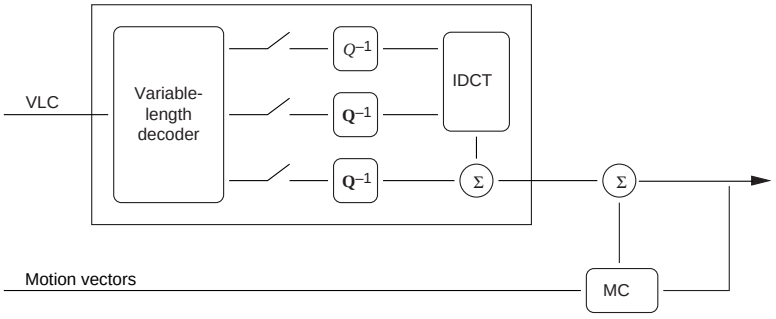

Figure 14. Video decoder based on residual vector quantization.

they are not intelligently encoded. The bit map at the first stage is combined with the motion vector information so that the number of bits is reduced. If the motion vector of a block is nonzero, it is assumed that the block needs to be encoded by at least the first stage; thus, the overhead first-stage flag bit for blocks with nonzero motion vectors is eliminated. The bit budget for these bit maps can be further reduced by the use of correlation between the second- and third-stage bit maps. Implementation details of an RVQ-based video codec are given by Kwon et al [14].

4.10.2 Performance of an RVQ-Based Video Codec

Simulation results are given for an RVQ-based video encoder with the following parameters: the first-stage scalar quantizer (SQ) had 8 quantization levels, the codebook size of the second-stage VQ was 16, and the codebook size of the third-stage VQ was 128. The number of significant DCT coefficients used was 9. Simulation results are given for the encoder / decoder operating at three different very low bit rates. The results are compared with those obtained with the H.263 codec [15,16]. For the H.263 codec, all negotiable options were turned on (except for "PB frames"), so that we could make a reasonable comparison with the RVQ codec (the PB frame mode can be easily incorporated in the RVQ codec). In the H.263 codec, the quantization parameter (QP) for the first frame was set to its maximum value, so that the codec would give the best performance for the "intra-frame." Since we were interested in the steady-state characteristics and not the first few transient frames, the bits consumed in the first frame were not included in the bit-rate calculations. The sequences used were those of the popular "salesman" test sequence. Each of these sequences has 8-bit pixels, with frame size 144×176, and the frame rate was 10 frames per second. We evaluated performance by using the peak signal-to-noise ratio (PSNR) measurement to compare the two coding methods. The computation requirements for the RVQ codec include real-time DCT; these requirements are the same as those of H.263. The transform-domain VQ has a codebook of size 16; therefore, its computational complexity is small. Only about 10 percent (average over all the sequences tested) of the blocks were encoded by the last VQ stage.

Forty motion-compensated difference frames extracted from four different sequences (in which the test sequences were not included) were used to train the codebooks. The codebooks were trained in a three-step process. We first designed the scalar quantizer using the Lloyd's algorithm. We then generated the second- and third-stage initial codebooks using the *k*-means algorithm. Finally, we retrained the three quantizers using the entropy constraint in a closed-loop manner to improve the rate-distortion performance.

The rate-distortion performance of the two codecs is shown in figure 15(a), which demonstrates that the RVQ codec outperformed H.263 at all three bit rates. At very low bit rates, the PSNR for H.263 decreases drastically, while that of the RVQ codec decreases gradually. The performance on the H.263 codec for the salesman sequence decreases dramatically at very low bit rates, because of the rather large motion in this sequence. In contrast, the RVQ codec handles such sequences very well at very low bit rates. Figures 15(b), (c), and (d) show the PSNR results of each reconstructed frame at three different bit rates. Since no rate control is used, a variable-bit-rate (VBR) bit stream with constant quality is generated. Figures 16(b), 17(b), and 18(b) show the reconstructed 48th frames for the salesman sequence compressed at the three different bit rates by the RVQ encoder. Figures 16(d), 17(d), and 18(d) show the reconstructed frames compressed by the H.263 encoder at the same bit rates. It can be clearly seen (especially at 5.4 kb/s) that H.263 suffers from blocking and smoothing, while the output of the RVQ codec is of much better visual quality.

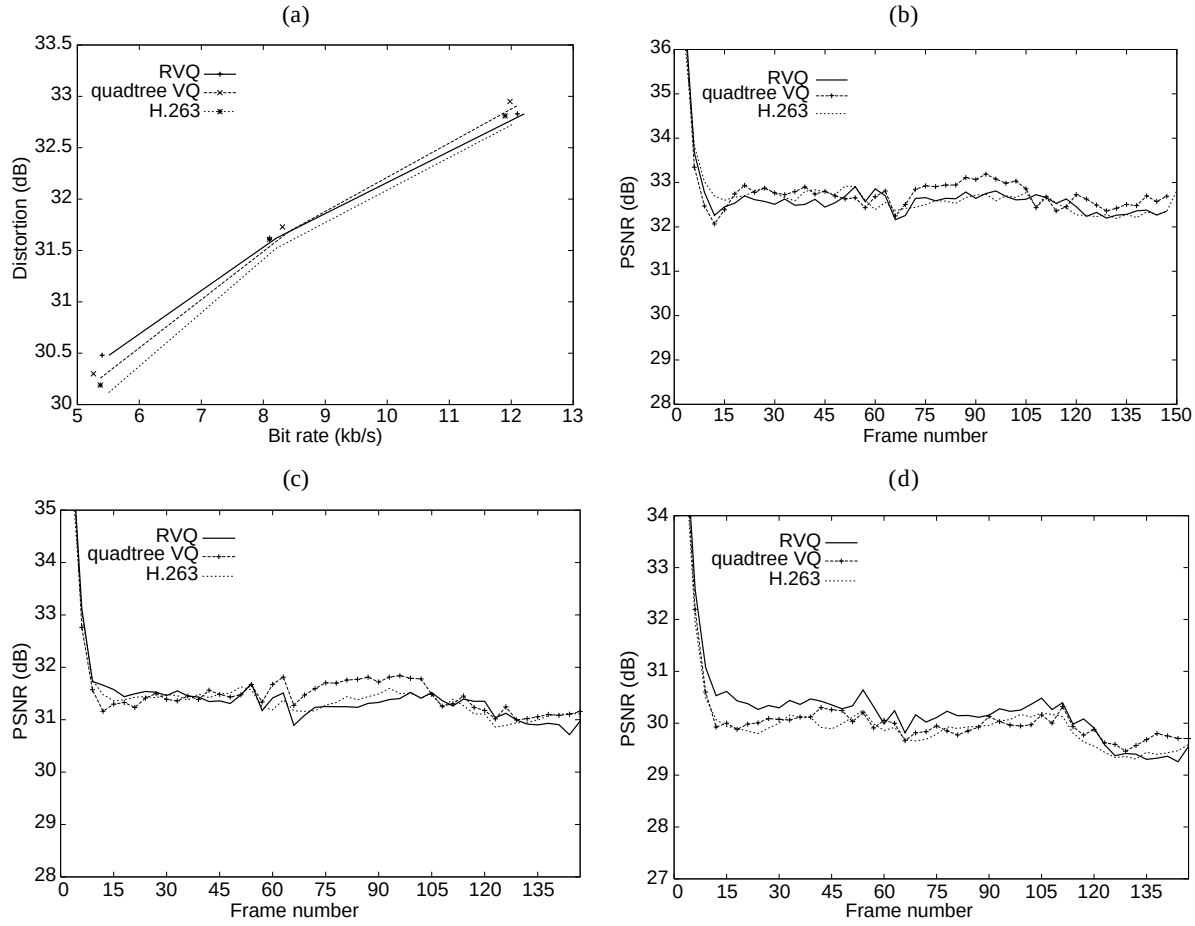

Figure 15. Performance of video compression algorithms using vector quantization: (a) rate-distortion performance, (b) PSNR results at 12 kb/s, (c) PSNR results at 8.1 kb/s, and (d) PSNR results at 5.3 kb/s.

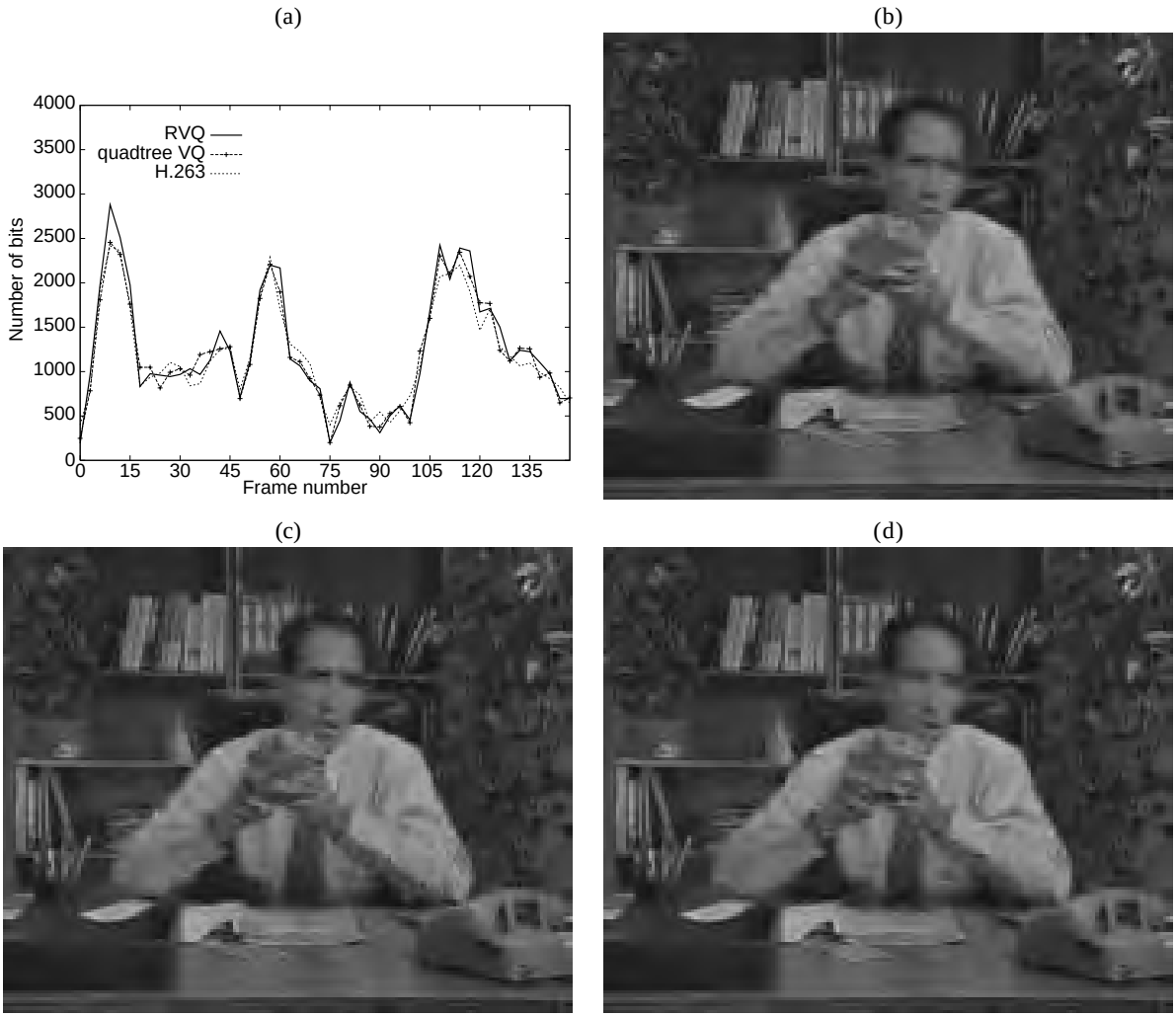

Figure 16. Results for bit rate of approximately 12 kb / s: (a) frame bit rate for sequence, (b) RVQ codec, (c) quadtree-VQ codec, and (d) H.263 codec.

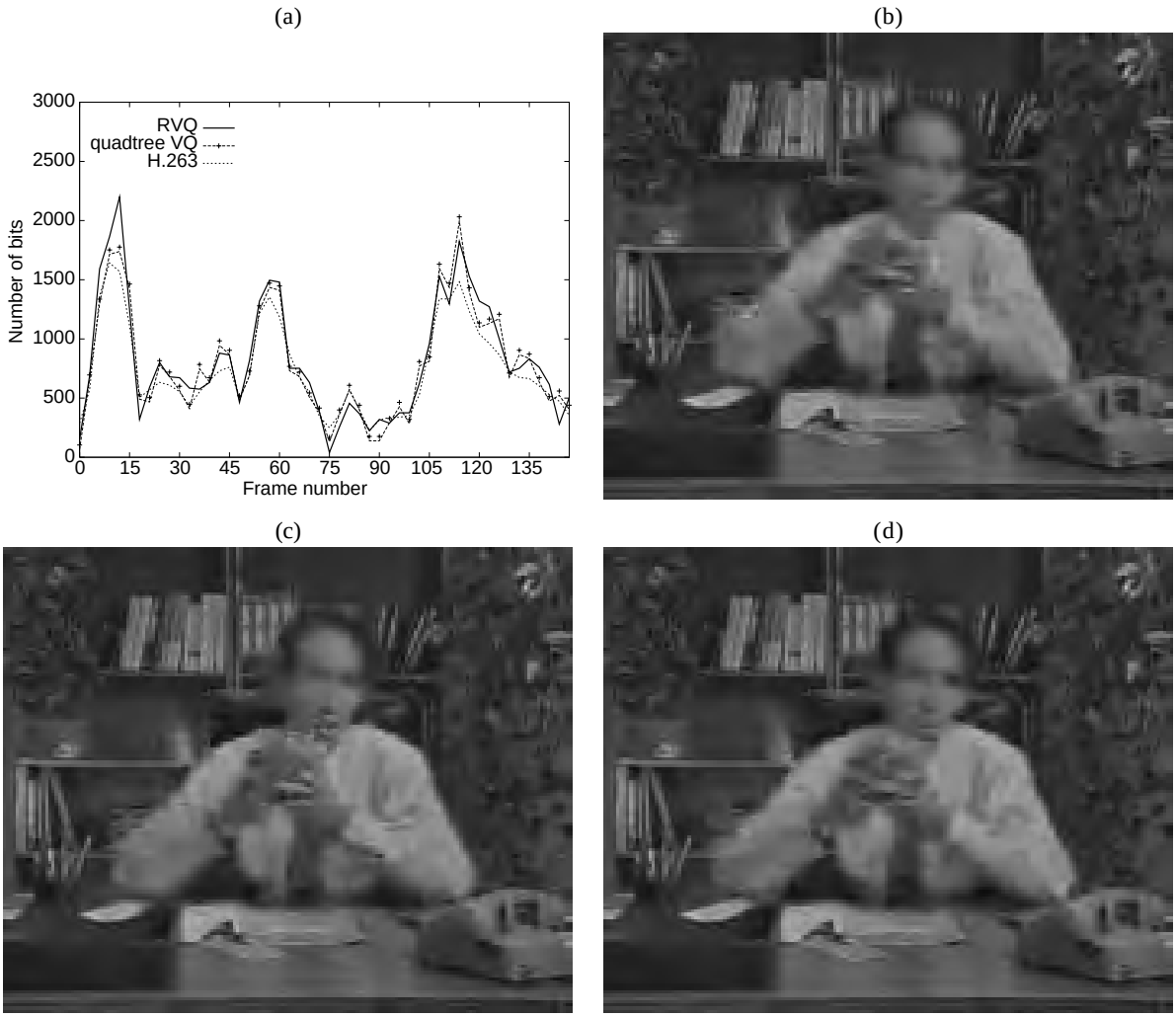

Figure 17. Results for bit rate of approximately 8.1 kb / s: (a) frame bit rate for sequence, (b) RVQ codec, (c) quadtree-VQ codec, and (d) H.263 codec.

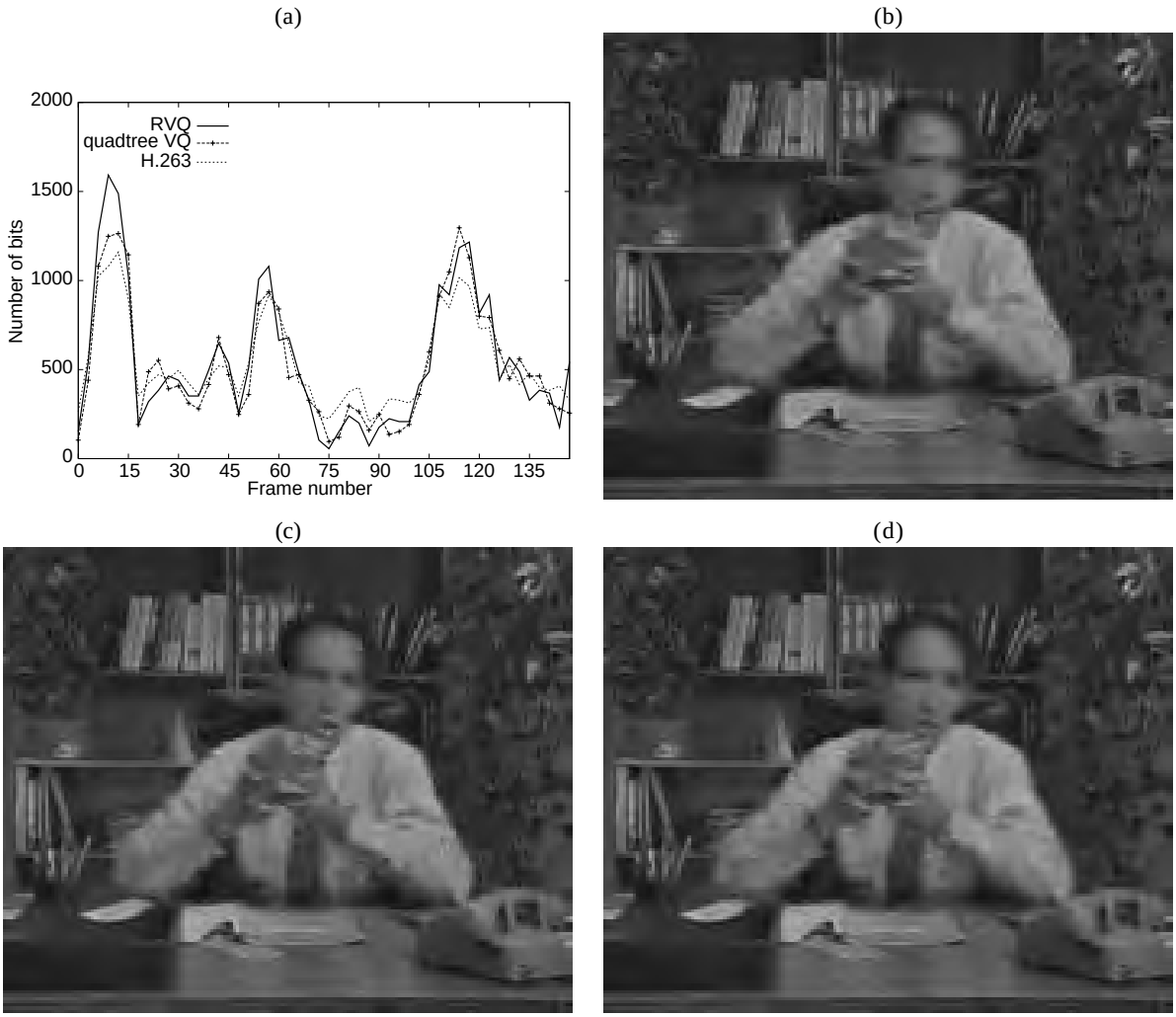

Figure 18. Results for bit rate of approximately 5.3 kb / s: (a) frame bit rate for sequence, (b) RVQ codec, (c) quadtree-VQ codec, and (d) H.263 codec.

5. Quadtree-Based Vector Quantization

A quadtree is a hierarchical data structure used to represent regions, curves, surfaces, and volumes. Representations of regions by a quadtree are achieved by the successive subdivision of the image array into four equal quadrants. This process is known as a regular decomposition of an array. An image is thus decomposed into homogeneous regions with sides of lengths that are powers of two. A tree of degree 4 (each nonleaf has four children) is generated to represent the image in terms of its homogeneous regions. The root node corresponds to the entire array, and each child of a node represents a quadrant of the region represented by that node. Leaf nodes of the tree correspond to those blocks for which no further subdivision is necessary. The above segmentation procedure is known as a top-down construction of the quadtree. Another possibility for constructing a quadtree is a bottom-up procedure, where small blocks are merged together recursively to form a larger block if they are homogeneous with respect to the merging criterion.

The regular decomposition method does not necessarily correspond to the segmentation of the image into maximal homogeneous regions. It is likely that unions of adjacent blocks form homogeneous regions. To obtain these maximal homogeneous regions, we must allow the merging of adjacent blocks. However, the resulting partition will no longer be represented by a quadtree; instead, the final representation is in the form of an adjacency graph. (An alternative method to obtain maximal homogeneous regions is to use a decomposition technique that is not regular: that is, it segments the image into rectangular blocks of arbitrary size. Such a method would require a different coding procedure for each block size.) Here, we use a regular decomposition method because the resulting blocks are square; this method reduces the complexity of the encoder, the decoder, and the number of bits required to represent the binary quadtree. The homogeneous regions so obtained are thus not necessarily maximal.

5.1 Quadtree Decomposition

A quadtree decomposition results in an unbalanced tree structure with leaf nodes of different sizes. In a regular decomposition, the leaf nodes are restricted to square blocks. It is further possible to restrict the sides of the leaf nodes to a small range of values. Such a restriction results in a tree

structure with the leaf nodes being square blocks with a maximum of n different sizes. A signal decomposed by the above method can be compressed by vector quantization of the leaf blocks. This will require n different VQs corresponding to the different block sizes. All leaf nodes of the same size are quantized by one VQ, as shown in figure 19.

The choice of criterion used in the quadtree decomposition is one of the most important factors in the design of a quadtree-based VQ.

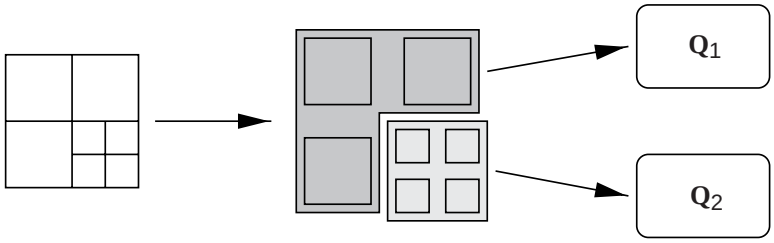

Figure 19. Vector quantization of quadtree leaf nodes.

5.2 Optimal Quadtree in the Rate-Distortion Sense

Quadtree decomposition for vector quantization can take into account the distortion introduced by the VQ. An optimal decomposition algorithm, in the rate-distortion sense, was introduced by Sullivan and Baker [17]. The algorithm attempts to minimize a constrained error function defined as follows:

$$\mathcal{E}^q = \mathbf{d}(\mathbf{X}, \mathbf{Q}^q(\mathbf{X})) + \lambda \cdot \mathbf{b}^q, \tag{18}$$

where $\mathbf{d}$ is the distortion introduced in quantizing the block k with a particular tree structure q, and $\mathbf{b}^q$ is the total number of bits used to represent the tree and the quantization indices. The Lagrangian λ controls the trade-off between the bit rate and distortion; it determines the operating point on the $\mathbf{R}$-$\mathbf{D}$ curve, as explained in section 4.8.2.

Consider a block $\mathbf{X}^m$ of size $2^m \times 2^m$ and its descendents $\mathbf{X}_i^{m-1}, i = 1 \ldots 4$. Let the distortion of quantizing the blocks $\mathbf{X}_i^{m-1}$ with the optimal quantizer be $\mathbf{d}_i^{m-1}$, and the number of bits necessary to optimally quantize $\mathbf{X}_i^{m-1}$ be $\mathbf{b}_i^{m-1}$. Similarly, let the distortion of quantizing the block $\mathbf{X}^m$ be $\mathbf{d}^m$, and the number of bits be $\mathbf{b}^m$. The four blocks $\mathbf{X}^{m-1}$ are merged and coded as a single block of size $2^m \times 2^m$ if the following condition is true:

$$\mathbf{d}^m + \lambda \mathbf{b}^m \leq \sum_i \mathbf{d}_i^{m-1} + \lambda \sum_i \mathbf{b}_i^{m-1}. \tag{19}$$

The above criterion can be used to prune the tree in a bottom-up manner to obtain the $\mathbf{R}$-$\mathbf{D}$ optimized hierarchical quantization scheme.

5.3 Video Compression Using Quadtree-Based Vector Quantization

The residual signal generated by the motion-compensation algorithm can be compressed by the quadtree VQ [18]. The residual signal $\mathbf{r}^0(i,j,t)$ is divided into blocks of size $2^m \times 2^m$. Each of these blocks is encoded by the **R-D** optimized, quadtree-based VQs. A **k**-stage hierarchical VQ uses **k** VQs, which work blocks of size $2^n \times 2^n, n = m, \ldots, m-k$. The quadtree bitmap is encoded as shown in figure 20.

We have implemented a video compression system using the quadtree VQ that we describe here. A simulation framework similar to that used in the RVQ video codec was used in evaluating the performance of the quadtree-VQ-based video compression algorithm. The residual signal after motion compensation was compressed by a three-stage quadtree VQ. The three quantizers used blocks of size 16×16, 8×8, and 4×4, respectively. Results are given for an encoder that uses scalar quantizers for blocks of size 16×16 and 8×8. Blocks of 4×4 are compressed by a VQ trained by the GLA algorithm. The quadtree was segmented with the **R-D** optimized algorithm. We tested the performance using the "salesman" sequence at three very low bit rates, as we did for the motion-compensated RVQ video codec. The performance of the quadtree-VQ compression algorithm was numerically similar to that of the RVQ compression algorithm. Figure 15 shows the PSNR results of each reconstructed frame at three different bit rates. Figures 16c, 17c, and 18c show the reconstructed 48th frames for the salesman sequence compressed at the three different bit rates by the quadtree-VQ encoder. As these figures show, the performance of the quadtree-VQ encoder is similar to that of the RVQ encoder at all the bit rates.

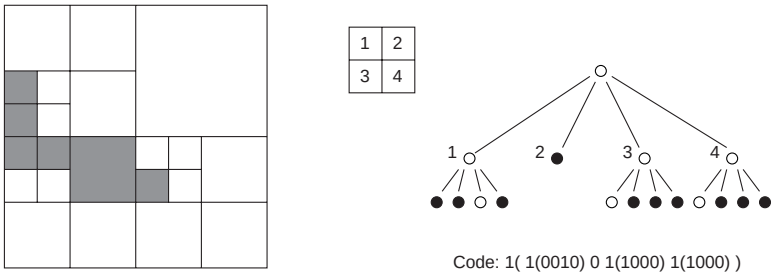

Figure 20. Encoding quadtree data structure.

6. Conclusions

In this report, we have explored two vector-quantization-based video compression algorithms. In our work we have identified two important areas that can be exploited to improve upon existing coding methods:

1. Multiscale segmentation: different areas in the image are coded at different scales. In vector quantization, this is equivalent to using different block sizes for different areas.
2. Multirate coding: different areas in the image are coded at different precisions, since all areas of the image do not contain the same amount of information.

We have used two different methods to achieve the above goals. In the RVQ-based encoder, we use the successive-refinement paradigm to achieve a variable rate. In the quadtree-VQ encoder, the rate variability is limited, but this technique is superior to a successive-refinement technique because it performs direct quantization. Both algorithms use variable block sizes. The resulting performance of these two encoders is similar, and both types are superior to existing video compression standards.

References

1. Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Trans. Commun.*, **28**(1):84–95, January 1980.

2. T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, 1984.

3. N. M. Nasrabadi and Y. Feng. Vector quantization of images based upon the Kohonen self-organization feature maps. *International Conference on Neural Networks*, **I**:101–105, 1988.

4. P. A. Chou, T. Lookabaugh, and R. M. Gray. Entropy-constrained vector quantization. *IEEE Trans. Acoust. Speech Signal Process.*, **37**(1), January 1989.

5. S. Roucos, J. Makhoul, and H. Gish. Vector quantization in speech coding. *Proc. IEEE*, **73**(11):1551–1587, November 1985.

6. B. H. Juang and A. H. Gray. Multiple stage vector quantization for speech coding. *Proc. IEEE ICASSP*, **1**:597–600, April 1982.

7. C. F. Barnes and R. L. Frost. Vector quantizers with direct sum codebooks. *IEEE Trans. Info. Theory*, **39**(2):565–580, March 1993.

8. W.H.R. Equitz and T. M. Cover. Successive refinement of information. *IEEE Trans. Inf. Theory*, **37**(2):269–275, March 1991.

9. C. F. Barnes, S. A. Rizvi, and N. M. Nasrabadi. Advances in residual vector quantization: A review. *IEEE Trans. Image Process.*, **5**(2), February 1996.

10. S. Gupta, W. Y. Chan, and A. Gersho. Enhanced multistage vector quantizer by joint codebook design. *IEEE Trans. Commun.*, **40**(11):1693–1697, November 1992.

11. S. A. Rizvi and N. M. Nasrabadi. Residual vector quantization using a multi-layer competitive neural network. *IEEE J. Selected Areas Commun.*, **12**(9):1452–1459, December 1994.

12. R. A. King and N. M. Nasrabadi. Image coding using vector quantization in the transform domain. *Pattern Recognition Lett.*, **1**:323–329, 1983.

13. I. H. Witten, R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Commun. ACM*, **30**(6):520–540, June 1987.

14. H. Kwon, M. Venkatraman, and N. M. Nasrabadi. Very low bit-rate video coding using variable block-size entropy-constrained residual vector quantizers. *IEEE J. Selected Areas Commun.*, **15**(9):1714–1725, December 1997.

15. K. N. Ngan, D. Chai, and A. Millin. Very low bit rate video coding using H.263 coder. *IEEE Trans. Circuits Syst. Video Technol.*, **6**(3):308–312, June 1996.

16. Telenor R&D. TMN (H.263) encoder/decoder, ftp://bonde.nta.no/pub/tmn. TMN codec, May 1996.

17. G. J. Sullivan and R. L. Baker. Efficient quadtree coding of images and video. *IEEE Trans. Image Process.*, **3**(3):327–331, May 1994.

18. N. M. Nasrabadi, S. Lin, and Y. Feng. Interframe hierarchical vector quantization. *Opt. Engineer.*, **28**(7):717–725, July 1989.

Distribution

Admnstr
Defns Techl Info Ctr
Attn DTIC-OCP
8725 John J Kingman Rd Ste 0944
FT Belvoir VA 22060-6218

Ofc of the Dir Rsrch and Engrg
Attn R Menz
Pentagon Rm 3E1089
Washington DC 20301-3080

Ofc of the Secy of Defns
Attn ODDRE (R&AT) G Singley
Attn ODDRE (R&AT) S Gontarek
The Pentagon
Washington DC 20301-3080

OSD
Attn OUSD(A&T)/ODDDR&E(R) R Tru
Washington DC 20301-7100

CECOM
Attn PM GPS COL S Young
FT Monmouth NJ 07703

CECOM
Sp & Terrestrial Commctn Div
Attn AMSEL-RD-ST-MC-M H Soicher
FT Monmouth NJ 07703-5203

Dept of the Army (OASA) RDA
Attn SARD-PT R Saunders
103 Army
Washington DC 20301-0103

Dir for MANPRINT
Ofc of the Deputy Chief of Staff for Prsnnl
Attn J Hiller
The Pentagon Rm 2C733
Washington DC 20301-0300

Hdqtrs Dept of the Army
Attn DAMO-FDT D Schmidt
400 Army Pentagon Rm 3C514
Washington DC 20301-0460

MICOM RDEC
Attn AMSMI-RD W C McCorkle
Redstone Arsenal AL 35898-5240

US Army ATCOM
Attn AMSAC-R-TD-CC R Wall
FT Eustis VA 23604-1104

US Army CECOM
Attn AMSEL-RD-C3-AC P Sass
FT Monmouth NJ 07703

US Army CECOM
Attn AVESD Gard
D II
FT Belvoir VA 22060-5806

US Army CECOM Rsrch, Dev, & Engrg Ctr
Attn R F Giordano
FT Monmouth NJ 07703-5201

Night Vsn & Elect Sensors Dirctrt
US Army Commctn-Elect Cmnd
Attn T Jones
10221 Burbeck Rd Ste 430
FT Belvoir VA 22060-5806

US Army Edgewood Rsrch, Dev, & Engrg Ctr
Attn SCBRD-TD J Vervier
Aberdeen Proving Ground MD 21010-5423

US Army Info Sys Engrg Cmnd
Attn ASQB-OTD F Jenia
FT Huachuca AZ 85613-5300

US Army Materiel Sys Analysis Agency
Attn AMXSY-D J McCarthy
Aberdeen Proving Ground MD 21005-5071

US Army Matl Cmnd
Dpty CG for RDE Hdqtrs
Attn AMCRD BG Beauchamp
5001 Eisenhower Ave
Alexandria VA 22333-0001

US Army Matl Cmnd Prin
Dpty for Acquisition Hdqrts
Attn AMCDCG-A D Adams
5001 Eisenhower Ave
Alexandria VA 22333-0001

Distribution (cont'd)

US Army Matl Cmnd
Prin Dpty for Techlgy Hdqrts
Attn AMCDCG-T M Fisette
5001 Eisenhower Ave
Alexandria VA 22333-0001

US Army Natick Rsrch, Dev, & Engrg Ct
Acting Techl Dir
Attn SSCNC-T P Brandler
Natick MA 01760-5002

US Army Rsrch Ofc
Attn AMXRO-EL W Sander
PO Box 12211
Research Triangle Park NC 22709-2211

US Army Rsrch Ofc
Attn G Iafrate
4300 S Miami Blvd
Research Triangle Park NC 27709

US Army Simulation, Train, & Instrmntn Cmnd
Attn J Stahl
12350 Research Parkway
Orlando FL 32826-3726

US Army TACOM
Attn AMSTA-TA-D P Hanjack MS 201B
Attn AMSTA-TR-R K Adams MS 264
Warren MI 48397

US Army Tank-Automtv & Armaments Cmnd
Attn AMSTA-AR-TD C Spinelli
Bldg 1
Picatinny Arsenal NJ 07806-5000

US Army Tank-Automtv Cmnd Rsrch, Dev, & Engrg Ctr
Attn AMSTA-TA J Chapin
Warren MI 48397-5000

US Army Test & Eval Cmnd
Attn R G Pollard III
Aberdeen Proving Ground MD 21005-5055

US Army Train & Doctrine Cmnd
Battle Lab Integration & Techl Dirctrt
Attn ATCD-B J A Klevecz
FT Monroe VA 23651-5850

US Military Academy
Dept of Mathematical Sci
Attn MAJ D Engen
West Point NY 10996

Nav Air Warfare Ctr
Attn S Gattis
1 Admin Circle
China Lake CA 93555-6001

Dir
Nav Rsrch Lab
Washington DC 20375-5000

Nav Surface Warfare Ctr
Attn Code B07 J Pennella
17320 Dahlgren Rd Bldg 1470 Rm 1101
Dahlgren VA 22448-5100

Comdt
US Nav Academy
Annapolis MD 21404

GPS Joint Prog Ofc Dir
Attn COL J Clay
2435 Vela Way Ste 1613
Los Angeles AFB CA 90245-5500

Comdt
US AF Academy
Colorado Springs CO 80840

DARPA
Attn B Kaspar
Attn L Stotts
3701 N Fairfax Dr
Arlington VA 22203-1714

ARL Electromag Group
Attn Campus Mail Code F0250 A Tucker
University of Texas
Austin TX 78712

Univ of Delaware Dept of Elect Engrg
Attn C Boncelet Jr
Newark DE 19716

Ericsson Inc Adv Dev & Rsrch Grp
Attn A Khayrallah
Research Triangle Park NC 27709

Distribution (cont'd)

ERIM
Attn J Ackenhusen
1975 Green Rd
Ann Arbor MI 48105

Natl Inst Standards/Tech
Attn J Phillips
Bldg 225 Rm A216
Gaithersburg MD 20899

Sanders Lockheed Martin Co
Attn PTP2-A001 K Damour
PO Box 868
Nashua NH 03061-0868

US Army Rsrch Lab
Attn AMSRL-IS-C R Kaste
Aberdeen Proving Ground MD 21005

US Army Rsrch Lab
Attn AMSRL-IS-CS A Mark
Attn AMSRL-CI-AC Angelni
Aberdeen Proving Ground MD 21005-5055

US Army Rsrch Lab
Attn AMSRL-CI-LL Techl Lib (3 copies)
Attn AMSRL-CS-AL-TA Mail & Records Mgmt
Attn AMSRL-CS-AL-TP Techl Pub (3 copies)
Attn AMSRL-IS P Emmerman
Attn AMSRL-IS R Slife
Attn AMSRL-IS-CI B Broome
Attn AMSRL-IS-CI T Mills
Attn AMSRL-IS-TA J Gowens
Attn AMSRL-IS-TP A Brodeen
Attn AMSRL-IS-TP A Downs

US Army Rsrch Lab (cont'd)
Attn AMSRL-IS-TP B Cooper
Attn AMSRL-IS-TP C Sarafidis
Attn AMSRL-IS-TP D Gwyn
Attn AMSRL-IS-TP D Torrieri
Attn AMSRL-IS-TP F Brundick
Attn AMSRL-IS-TP G Cirincione
Attn AMSRL-IS-TP G Hartwig
Attn AMSRL-IS-TP H Caton
Attn AMSRL-IS-TP L Wrencher
Attn AMSRL-IS-TP M Lopez
Attn AMSRL-IS-TP M Markowski
Attn AMSRL-IS-TP M Retter
Attn AMSRL-IS-TP S Chamberlain
Attn AMSRL-SC-I A Raglin
Attn AMSRL-SC-I T Hanratty
Attn AMSRL-SC-SA Wall
Attn AMSRL-SE J M Miller
Attn AMSRL-SE J Pellegrino
Attn AMSRL-SE-EE Z G Sztankay
Attn AMSRL-SE-SE D Nguyen
Attn AMSRL-SE-SE H Kwon
Attn AMSRL-SE-SE L Bennett
Attn AMSRL-SE-SE M Venkatraman (5 copies)
Attn AMSRL-SE-SE M Vrabel
Attn AMSRL-SE-SE N Nasrabadi
Attn AMSRL-SE-SE P Rauss
Attn AMSRL-SE-SE S Der
Attn AMSRL-SE-SE T Kipp
Attn AMSRL-SE-SR A M P Marinelli
Attn AMSRL-SE-SS V Marinelli
Adelphi MD 20783-1197

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 1998	Interim, May 1997 to July 1997

4. TITLE AND SUBTITLE
Video Compression using Vector Quantization

5. FUNDING NUMBERS
PE: 61102A

6. AUTHOR(S)
Mahesh Venkatraman, Heesung Kwon, and Nasser M. Nasrabadi

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Attn: AMSRL-SE-SE (e-mail: nnasraba@arl.mil)
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-1535

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
AMS code: 611102.305
ARL PR: 7NE0M1

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report presents some results and findings of our work on very-low-bit-rate video compression systems using vector quantization (VQ). We have identified multiscale segmentation and variable-rate coding as two important concepts whose effective use can lead to superior compression performance. Two VQ algorithms that attempt to use these two aspects are presented: one based on residual vector quantization and the other on quadtree vector quantization. Residual vector quantization is a successive approximation quantizer technique and is ideal for variable-rate coding. Quadtree vector quantization is inherently a multiscale coding method. The report presents the general theoretical formulation of these algorithms, as well as quantitative performance of sample implementations.

14. SUBJECT TERMS
Motion estimation, residual VQ, quadtree VQ

15. NUMBER OF PAGES
47

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

DEPARTMENT OF THE ARMY
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

An Equal Opportunity Employer